UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/796,613	03/08/2004	Leela S. Tamma	MS1-1880US	1791

22801 7590 05/08/2007

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
CHEN, QING	
ART UNIT	**PAPER NUMBER**
2191	

NOTIFICATION DATE	DELIVERY MODE
05/08/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lhptoms@leehayes.com

Office Action Summary	Application No.	Applicant(s)	
	10/796,613	TAMMA ET AL.	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 March 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on March 8, 2004.

2. **Claims 1-31** are pending.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- Reference number 406 in Figure 4.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not

accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The disclosure is objected to because of the following informalities:

- The specification contains the following typographical errors:
  - A whitespace character should be added between "version 1.2" and "are" on page 7, paragraph [0033].
  - A period (.) should be added after the paragraph on page 19, paragraph [0067].

Appropriate correction is required.

5. The use of trademarks, such as JAVA and CLEARCASE, has been noted in this application. Trademarks should be capitalized wherever they appear (capitalize each letter OR accompany each trademark with an appropriate designation symbol, *e.g.*, ™ or ®) and be accompanied by the generic terminology (use trademarks as adjectives modifying a descriptive noun, *e.g.*, "the JAVA programming language").

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner, which might adversely affect their validity as trademarks.

Claim Objections

6. **Claims 7, 10, 23, and 31** are objected to because of the following informalities:

- **Claim 7** contains a typographical error: "manipulate" should read -- manipulation --.
- **Claim 10** contains a typographical error: the word "and" should be added after the second-to-last limitation.
- **Claim 23** contains a typographical error: the article "A" in "A memory" and "A processor" should be changed to lowercase.
- **Claim 31** recites the limitation "the database." Applicant is advised to change this limitation to read "the relational database" for the purpose of providing it with proper explicit antecedent basis.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. **Claims 12, 20-22, and 29-31** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 12 and 29-31 recite computer-readable media as a claimed element. However, the limitation of "computer-readable media having/comprising computer-readable instructions" can be reasonably interpreted as the computer-readable media carrying or transmitting electrical signals, since the computer-readable instructions are not recorded on the computer-readable media, so as to permit the function of the descriptive material to be realized when executed.

Claims that recite nothing but the physical characteristics of a form of energy, such as a frequency, voltage, or the strength of a magnetic field, define energy or magnetism *per se*, and as such are non-statutory natural phenomena. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 112-14 (1853). Moreover, it does not appear that a claim reciting a signal encoded with functional descriptive material falls within any of the categories of patentable subject matter set forth in § 101.

Claims 20-22 are directed to systems. However, the recited components of the systems appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. The claims are directed to systems of functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. **Claims 1-6, 8, 12, and 23-29** are rejected under 35 U.S.C. 102(e) as being anticipated by **Hotti et al. (US 6,970,876)**.

As per **Claim 1**, Hotti et al. disclose:

- automatically determining a first set of data definition language (DDL) scripts associated with the particular version of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- automatically determining a second set of data manipulation language scripts associated with the particular version of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation*

Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts."); and

- generating an installation file comprising a union of the first set and the second set *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.").*

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Hotti et al. further disclose:

- wherein the particular version is associated with a first version in a sequence of one or more versions of the relational database *(see Column 2: 7-8, ""Schema revision" is a snapshot version of a schema that is identifiable by logical name or version number.").*

As per **Claim 3**, the rejection of **Claim 2** is incorporated; and Hotti et al. further disclose:

- wherein the automatically determining a first set comprises extracting a filename from metadata associated with the first version, the filename associated with a file comprising a data definition language script *(see Column 6: 63-66, "As part of the registration, the identification data, e.g. schema name, of the new application database node is sent to the configuration management master database node.").*

As per **Claim 4**, the rejection of **Claim 2** is incorporated; and Hotti et al. further disclose:

- wherein the automatically determining a second set comprises extracting a filename from metadata associated with the first version, the filename associated with a file comprising a data manipulation language script *(see Column 6: 63-66, "As part of the registration, the identification data, e.g. schema name, of the new application database node is sent to the configuration management master database node.").*

As per **Claim 5**, the rejection of **Claim 2** is incorporated; and Hotti et al. further disclose:

- wherein the generating an installation file comprises copying a data definition language script from a script file associated with the first set into the installation file *(see Column 7: 4-6, "Next the schema of the application master database node is created using the scripts that were downloaded to the new replica database node, 310.").*

As per **Claim 6**, the rejection of **Claim 2** is incorporated; and Hotti et al. further disclose:

- wherein the generating an installation file comprises copying a data manipulation language script from a script file associated with the second set into the installation file *(see Column 7: 4-6, "Next the schema of the application master database node is created using the scripts that were downloaded to the new replica database node, 310.").*

As per **Claim 8**, the rejection of **Claim 1** is incorporated; and Hotti et al. further disclose:

- wherein metadata exists that describes a sequence of multiple versions of the relational database where each version is an upgrade from a previous version, and the particular

version is not a first version in the sequence *(see Column 2: 7-8, ""Schema revision" is a snapshot version of a schema that is identifiable by logical name or version number." and 14-15, ""Schema script publication" is a system publication that contains the schema scripts of the database hierarchy.")*.

As per **Claim 12**, the rejection of **Claim 1** is incorporated; and Hotti et al. further disclose:

- One or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to implement the method as recited in claim 1 *(see Column 5: 1-5, "The invention also relates to a storage media comprising a stored, readable computer program, which is characterized in that the program comprises instructions for controlling a data management system or components thereof to implement the method according to the invention.")*.

As per **Claim 23**, Hotti et al. disclose:

- a memory *(see Column 3: 66-67, "A database system may include server computers, smart terminals, other terminals and network nodes.")*;

- a processor *(see Column 3: 66-67, "A database system may include server computers, smart terminals, other terminals and network nodes.")*; and

- a database schema version management system stored in the memory, executed on the processor, and configured to manage schema data associated with multiple versions of a relational database *(see Figure 2a: 234; Column 2: 7-8, ""Schema revision" is a snapshot*

version of a schema that is identifiable by logical name or version number."; Column 6: 17-20, "The configuration management node 231 includes a configuration management application 234 for managing the schemas and application configuration of the database system.").

As per **Claim 24**, the rejection of **Claim 23** is incorporated; and Hotti et al. further disclose:

- wherein the schema data identifies a script associated with a data definition language object of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.").*

As per **Claim 25**, the rejection of **Claim 23** is incorporated; and Hotti et al. further disclose:

- wherein the schema data identifies a script associated with a data manipulation language object of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.").*

As per **Claim 26**, the rejection of **Claim 23** is incorporated; and Hotti et al. further disclose:

- wherein the database schema version management system is further configured to generate an installation file associated with an initial version of the relational database *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.").*

As per **Claim 27**, the rejection of **Claim 23** is incorporated; and Hotti et al. further disclose:

- wherein the database schema version management system is further configured to generate an installation file associated with a non-initial version of the relational database *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.").*

As per **Claim 28**, the rejection of **Claim 23** is incorporated; and Hotti et al. further disclose:

- wherein the database schema version management system is further configured to generate an upgrade file for upgrading a first version of the relational database to another version of the relational database *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.")*.

As per **Claim 29**, Hotti et al. disclose:

- maintain schema data that identifies scripts associated with database objects of multiple sequential versions of a relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node." and 14-15, ""Schema script publication" is a system publication that contains the schema scripts of the database hierarchy."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*; and

- generate an installation file associated with an initial version of the relational database by applying laws of set theory to the schema data to identify scripts associated with the database objects of the initial version of the relational database *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the*

configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.").

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. **Claim 7** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Hotti et al. (US 6,970,876)** in view of **Orcacle8*i* Distributed Database Systems Release 8.1.5, 1999 (hereinafter Oracle1999)**.

As per **Claim 7**, the rejection of **Claim 6** is incorporated; however, Hotti et al. do not disclose:

- wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file.

Oracle1999 discloses:

- wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file *(see Page 2-13)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Oracle1999 into the teaching of Hotti et al. to include wherein the copying further comprises prepending a create command to the data manipulation language script in the installation file. The modification would be obvious because one of ordinary skill in the art would be motivated to define a stored procedure.

13. **Claim 9** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Hotti et al. (US 6,970,876)** in view of **Carey et al. (US 6,947,945)**.

As per **Claim 9**, the rejection of **Claim 8** is incorporated; however, Hotti et al. do not disclose:

- wherein the metadata comprises an XML file.

Carey et al. disclose:

- wherein the metadata comprises an XML file *(see Column 1: 30-32, "An alternative data format to the tables found in an RDBMS is XML, which is a tag language for describing documents.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey et al. into the teaching of Hotti et al. to include wherein the metadata comprises an XML file. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize a future standard for information exchange between peer data stores, and between client visualization tools and data servers *(see Carey et al. – Column 2: 15-17)*.

14. **Claims 10, 11, 13-20, 30, and 31** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hotti et al. (US 6,970,876)** in view of **Baisley et al. (US 6,415,299)**.

As per **Claim 10**, the rejection of **Claim 8** is incorporated; however, Hotti et al. do not disclose:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j; and

- determining the first set as the union of sets $A_1, A_2, ..., A_j$.

Baisley et al. disclose:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version *(see Figure 3; Column 5: 17-36, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model.")*;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*; and

- determining the first set as the union of sets $A_1, A_2, ..., A_j$ *(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data definition language script associated with the first version;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each

associated with a file comprising a data definition language script to be executed when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j; and

- determining the first set as the union of sets $A_1, A_2, ..., A_j$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24)*.

As per **Claim 11**, the rejection of **Claim 8** is incorporated; however, Hotti et al. do not disclose:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j;

- iteratively extracting a set B_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when

upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j; and

- determining the second set C_j by determining:

$$C_2 = [A_1 \bigcup A_2] - B_2,$$
$$C_3 = [C_2 \bigcup A_3] - B_3,$$
$$C_4 = [C_3 \bigcup A_4] - B_4,$$
$$\ldots$$
$$C_j = [C_{j-1} \bigcup A_j] - B_j.$$

Baisley et al. disclose:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version *(see Figure 3; Column 5: 17-36, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model.")*;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*;

- iteratively extracting a set B_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a script to be executed when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*; and

- determining the second set C_j by determining:

$$C_2 = [A_1 \bigcup A_2] - B_2,$$
$$C_3 = [C_2 \bigcup A_3] - B_3,$$
$$C_4 = [C_3 \bigcup A_4] - B_4,$$
$$\ldots$$
$$C_j = [C_{j-1} \bigcup A_j] - B_j$$

(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.").

Official Notice is taken that it is old and well known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- extracting a set A_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

- iteratively extracting a set A_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j;

- iteratively extracting a set B_i comprising zero or more filenames from metadata associated with an i^{th} version of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version $i-1$ of the relational database to version i of the relational database, where i varies incrementally from 2 to j, where the particular version is j; and

- determining the second set C_j by determining:

$$C_2 = [A_1 \bigcup A_2] - B_2,$$
$$C_3 = [C_2 \bigcup A_3] - B_3,$$
$$C_4 = [C_3 \bigcup A_4] - B_4,$$
$$\ldots$$
$$C_j = [C_{j-1} \bigcup A_j] - B_j.$$

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 13**, Hotti et al. disclose:

- determining a set *A* of data definition language (DDL) scripts that, when executed, perform creates, alters, and drops of DDL objects associated with version *i* of the relational database, resulting in DDL objects associated with version *j* of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- determining a set *B* of data manipulation language (DML) scripts that, when executed, create DML objects that are associated with version *j* of the relational database, but that are not associated with version *i* of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- determining a set *C* of DML scripts that, when executed, modify DML objects that are associated with both version *i* and version *j* of the relational database, but that differ between version *i* and version *j* of the relational database *(see Column 6: 30-37, "The updating*

between the configuration management master and the configuration management replicas can be made using the synchronization functionality of the servers [1]."); and

- determining a set D of scripts that are associated with version i of the relational database, but that are not associated with version j of the relational database *(see Column 6: 30-37, "The updating between the configuration management master and the configuration management replicas can be made using the synchronization functionality of the servers [1]."*).

However, Hotti et al. do not disclose:

- generating an upgrade file comprising a union of sets A, B, C, and D $(A \bigcup B \bigcup C \bigcup D)$.

Baisley et al. disclose:

- generating an upgrade file comprising a union of sets A, B, C, and D $(A \bigcup B \bigcup C \bigcup D)$ *(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists."*).

Official Notice is taken that it is old and well known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- determining a set A of data definition language (DDL) scripts that, when executed, perform creates, alters, and drops of DDL objects associated with version i of the relational database, resulting in DDL objects associated with version j of the relational database;

- determining a set B of data manipulation language (DML) scripts that, when executed, create DML objects that are associated with version j of the relational database, but that are not associated with version i of the relational database;

- determining a set C of DML scripts that, when executed, modify DML objects that are associated with both version i and version j of the relational database, but that differ between version i and version j of the relational database;

- determining a set D of DML drop scripts that, when executed, drop DML objects that are associated with version i of the relational database, but that are not associated with version j of the relational database; and

- generating an upgrade file comprising a union of sets A, B, C, and D $(A \bigcup B \bigcup C \bigcup D)$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 14**, the rejection of **Claim 13** is incorporated; however, Hotti et al. do not disclose:

- iteratively extracting sets M_k, each comprising zero or more filenames from metadata associated with a k^{th} version of the relational database, where $i < k <= j$, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version $k-1$ of the relational database to version k of the relational database; and

- determining the set A as the union of sets $M_{i+1}, M_{i+2}, ..., M_j$ $(A = M_{i+1} \bigcup M_{i+2} \bigcup ... \bigcup M_j)$.

Baisley et al. disclose:

- iteratively extracting sets M_k, each comprising zero or more filenames from metadata associated with a k^{th} version of the relational database, where $i < k <= j$, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version $k-1$ of the relational database to version k of the relational database *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*; and

- determining the set A as the union of sets $M_{i+1}, M_{i+2}, ..., M_j$ $(A = M_{i+1} \bigcup M_{i+2} \bigcup ... \bigcup M_j)$ *(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version;*

and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- iteratively extracting sets M_k, each comprising zero or more filenames from metadata associated with a k^{th} version of the relational database, where $i < k <= j$, the zero or more filenames each associated with a file comprising a data definition language script to be executed when upgrading from version $k-1$ of the relational database to version k of the relational database; and

- determining the set A as the union of sets $M_{i+1}, M_{i+2}, ..., M_j$ $(A = M_{i+1} \bigcup M_{i+2} \bigcup ... \bigcup M_j)$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 15**, the rejection of **Claim 13** is incorporated; however, Hotti et al. do not disclose:

- determining a set E of DML scripts that when executed:

- perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

- perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database; and

- determining set B as the difference between sets E and C $(B = E - C)$.

Official Notice is taken that it is old and well known within the computing art to perform alters and creates of DML objects. A query language often provides "create" and "alter" commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley et al. disclose:

- determining set B as the difference between sets E and C $(B = E - C)$ *(see Column 5: 47-67, "TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- determining a set E of DML scripts that when executed:

- perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

- perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database; and

- determining set B as the difference between sets E and C $(B = E - C)$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 16**, the rejection of **Claim 15** is incorporated; however, Hotti et al. do not disclose:

- iteratively determining a set P_x of DML scripts that when executed will upgrade DML objects from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j;

- iteratively determining a set N_x of DML scripts that when executed will drop DML objects that are associated with version $x-1$ of the relational database but that are not associated with version x of the relational database, where x varies incrementally from $i+2$ to j;

- iteratively determining a set M_x of DML scripts that when executed will upgrade DML objects from version i of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j, and where:

$$M_{i+1} = P_{i+1}$$
$$M_{i+2} = [M_{i+1} \bigcup P_{i+2}] - N_{1+2}$$
$$M_{i+3} = [M_{i+2} \bigcup P_{i+3}] - N_{i+3}$$
...
$$M_j = [M_{j-1} \bigcup P_j] - N_j \text{; and}$$

determining set $E = M_j$.

Baisley et al. disclose:

- iteratively determining a set P_x of DML scripts that when executed will upgrade DML objects from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*;

- iteratively determining a set N_x of DML scripts that are associated with version $x-1$ of the relational database but that are not associated with version x of the relational database, where x varies incrementally from $i+2$ to j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*;

- iteratively determining a set M_x of DML scripts that when executed will upgrade DML objects from version i of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j, and where:

$$M_{i+1} = P_{i+1}$$
$$M_{i+2} = [M_{i+1} \bigcup P_{i+2}] - N_{1+2}$$
$$M_{i+3} = [M_{i+2} \bigcup P_{i+3}] - N_{i+3}$$
...
$$M_j = [M_{j-1} \bigcup P_j] - N_j$$

(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two."); and

determining set $E = M_j$ *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.").*

Official Notice is taken that it is old and well known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- iteratively determining a set P_x of DML scripts that when executed will upgrade DML objects from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j;

- iteratively determining a set N_x of DML scripts that when executed will drop DML objects that are associated with version $x-1$ of the relational database but that are not associated with version x of the relational database, where x varies incrementally from $i+2$ to j;

- iteratively determining a set M_x of DML scripts that when executed will upgrade DML objects from version i of the relational database to version x of the relational database, where x varies incrementally from $i+1$ to j, and where:

$$M_{i+1} = P_{i+1}$$
$$M_{i+2} = [M_{i+1} \bigcup P_{i+2}] - N_{1+2}$$
$$M_{i+3} = [M_{i+2} \bigcup P_{i+3}] - N_{i+3}$$
$$\ldots$$
$$M_j = [M_{j-1} \bigcup P_j] - N_j; \text{ and}$$

determining set $E = M_j$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 17**, the rejection of **Claim 13** is incorporated; and Hotti et al. further disclose:

- determining a set F_j of DML scripts that when executed, create DML objects associated with version j of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*.

However, Hotti et al. do not disclose:

- determining a set E of DML scripts that when executed:
  - perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and
  - perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database; and
- determining set C as the intersection of set E and set F_j $(C = E \bigcap F_j)$.

Official Notice is taken that it is old and well known within the computing art to perform alters and creates of DML objects. A query language often provides "create" and "alter" commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley et al. disclose:

- determining set C as the intersection of set E and set F_j $(C = E \bigcap F_j)$ *(see Column 5: 47-67, "TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- determining a set E of DML scripts that when executed:
  - perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and
  - perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database; and
- determining set C as the intersection of set E and set F_j $(C = E \bigcap F_j)$.

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 18**, the rejection of **Claim 17** is incorporated; however, Hotti et al. do not disclose:

- extracting a set M_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

- iteratively extracting a set M_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j;

- iteratively extracting a set B_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j; and

- determining the set F_j by determining:

$$F_2 = [M_1 \bigcup M_2] - B_2,$$
$$F_3 = [F_2 \bigcup M_3] - B_3,$$
$$F_4 = [F_3 \bigcup M_4] - B_4,$$
$$\ldots$$
$$F_j = [F_{j-1} \bigcup M_j] - B_j.$$

Baisley et al. disclose:

- extracting a set M_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data

manipulation language (DML) script associated with the first version *(see Figure 3; Column 5: 17-36, "We start with a first model version 30 (or V1) where an attribute A.X=0 of the model.")*;

- iteratively extracting a set M_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*;

- iteratively extracting a set B_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file when upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j *(see Figure 3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.")*; and

- determining the set F_j by determining:

$$F_2 = [M_1 \bigcup M_2] - B_2,$$
$$F_3 = [F_2 \bigcup M_3] - B_3,$$
$$F_4 = [F_3 \bigcup M_4] - B_4,$$
...
$$F_j = [F_{j-1} \bigcup M_j] - B_j$$

(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.").

Official Notice is taken that it is old and well known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a drop script. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- extracting a set M_1 comprising one or more filenames from metadata associated with a first version in the sequence, the one or more filenames associated with a file comprising a data manipulation language (DML) script associated with the first version;

- iteratively extracting a set M_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML script to be executed to add or modify a DML object when

upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j;

- iteratively extracting a set B_x comprising zero or more filenames from metadata associated with version x of the relational database, the zero or more filenames each associated with a file comprising a DML drop script to be executed to drop a DML object when upgrading from version $x-1$ of the relational database to version x of the relational database, where x varies incrementally from 2 to j; and

- determining the set F_j by determining:

$$F_2 = [M_1 \bigcup M_2] - B_2,$$
$$F_3 = [F_2 \bigcup M_3] - B_3,$$
$$F_4 = [F_3 \bigcup M_4] - B_4,$$
$$\ldots$$
$$F_j = [F_{j-1} \bigcup M_j] - B_j.$$

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 19**, the rejection of **Claim 13** is incorporated; however, Hotti et al. do not disclose:

- determining a set E of DML scripts that when executed:

- perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and

- perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database;

- iteratively determining a set F_x of DML scripts that when executed, drop DML objects associated with version $x-1$ of the relational database that are not associated with version x of the relational database, where x varies incrementally from $i+1$ to j;

- determining a set G as the union of sets $F_i, F_{i+1}, F_{i+2}, ..., F_j$ ($G = F_i \bigcup F_{i+1} \bigcup ... \bigcup F_{i+2}$); and

- determining set D as the difference between set G and set E ($D = G - E$).

Official Notice is taken that it is old and well known within the computing art to perform alters and creates of DML objects. A query language often provides "create" and "alter" commands to create and modify, respectively, a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include performing alters and creates of DML objects. The modification would be obvious because one of ordinary skill in the art would be motivated to create and/or modify a database object.

Baisley et al. disclose:

- iteratively determining a set F_x of DML scripts that when executed, drop DML objects associated with version $x-1$ of the relational database that are not associated with version x of the relational database, where x varies incrementally from $i+1$ to j *(see Figure*

3; Column 5: 17-36, "In the next model version 31 (or V2) the attribute A.X is still equal to zero. However in a new branch, model version 32 (or V2A) A.X is now set equal to 2 (A.X=2). Likewise, in model version 33 (or V2B) A.X is still equal to two.");

- determining a set G as the union of sets $F_i, F_{i+1}, F_{i+2}, ..., F_j$

($G = F_i \bigcup F_{i+1} \bigcup ... \bigcup F_{i+2}$) *(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.")*; and

- determining set D as the difference between set G and set E ($D = G - E$) *(see Column 5: 47-67, "TABLE I below illustrates the conflict resolution, which may be made between versions 33 and 34.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include:

- determining a set E of DML scripts that when executed:
  - perform alters of DML objects associated with version i and version j of the relational database, but that differ between version i and version j of the relational database; and
  - perform creates of DML objects that are associated with version j of the relational database but that are not associated with version i of the relational database;

- iteratively determining a set F_x of DML scripts that when executed, drop DML objects associated with version $x-1$ of the relational database that are not associated with version x of the relational database, where x varies incrementally from $i+1$ to j;

- determining a set G as the union of sets $F_i, F_{i+1}, F_{i+2}, ..., F_j$ ($G = F_i \bigcup F_{i+1} \bigcup ... \bigcup F_{i+2}$); and

- determining set D as the difference between set G and set E ($D = G - E$).

The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 20**, Hotti et al. disclose:

- one or more data definition language (DDL) scripts, each associated with one or more versions of a relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- one or more data manipulation language (DML) scripts, each associated with one or more versions of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- a database schema version management structure definition *(see Column 1: 39-43, ""Database Schema" is the structure of a database system, described in a formal language supported by the database management system (DBMS). In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.")*;

- schema data associated with multiple versions of the relational database, the schema data organized according to the database schema version management structure definition *(see Column 2: 7-8, ""Schema revision" is a snapshot version of a schema that is identifiable by logical name or version number." and 14-15, ""Schema script publication" is a system publication that contains the schema scripts of the database hierarchy.")*; and

- an installation file generator configured to generate a file comprising the one or more DDL scripts associated with a particular one of the multiple versions of the relational database, and the one or more DML scripts associated with the particular one of the multiple versions of the relational database *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.")*.

However, Hotti et al. do not disclose:

- applying laws of set theory to the schema data.

Baisley et al. disclose:

- applying laws of set theory to the schema data *(see Column 2: 14-19, "The method comprises the steps of building a first list as a collection of versions that occur only in a history of the source version; and, building a second list as a collection of versions that occur only in a history of the target version. Next, a dual history is created as a union of the first and second lists.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Baisley et al. into the teaching of Hotti et al. to include applying laws of set theory to the schema data. The modification would be obvious because one of ordinary skill in the art would be motivated to define track and maintain objects, models and versions in an object oriented repository *(see Baisley et al. – Column 1: 21-24).*

As per **Claim 30**, the rejection of **Claim 29** is incorporated; and Hotti et al. further disclose:

- generate an installation file associated with a non-initial version of the relational database by applying laws of set theory to the schema data to identify *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server."):*

- scripts associated with data definition language (DDL) objects that are associated with the non-initial version of the relational database *(see Column 2: 9-10, ""Schema script" is a*

script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts."); and

- scripts associated with data manipulation language (DML) objects that are associated with the non-initial version of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*.

As per **Claim 31**, the rejection of **Claim 29** is incorporated; and Hotti et al. further disclose:

- generate an upgrade file associated with an upgrade from a first, but not necessarily initial, version of the relational database to a second, later, but not necessarily immediately sequential, version of the relational database by applying laws of set theory to the schema data to identify *(see Column 3: 21-25, "These synchronized schema/application configuration management replicas comprise scripts that are used for creating and/or updating the schemas of the database nodes and managing the configurations of applications that use the database node."; Column 7: 1-4, "This downloads the schema creation scripts and possibly also application configuration data such as software binaries and installation programs of the application master to the database server.")*:

- data definition language (DDL) scripts associated with DDL objects of the relational database that have been created or modified between the first and second versions of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*;

- data manipulation language (DML) scripts associated with DML objects of the relational database that have been created between the first and second versions of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*; and

- DML scripts associated with DML objects of the relational database that have been modified between the first and second versions of the relational database *(see Column 2: 9-10, ""Schema script" is a script that creates a schema or creates a new revision of an existing schema of a database node."; Column 4: 49-58, "Schema scripts can also include DML (Data Manipulation Language) or DDL (Data Definition Language) scripts, or any other data manipulation scripts.")*.

However, Hotti et al. do not disclose:

- drop scripts associated with database objects that have been dropped and not re-created between the first and second versions of the relational database.

Official Notice is taken that it is old and well known within the computing art to include a drop script as part of the DML. A query language often provides a "drop" command to remove a database object. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include drop scripts associated with database objects that have been dropped and not re-created between the first and second versions of the relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to remove a database object.

15. **Claims 21 and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hotti et al. (US 6,970,876)** in view of **Baisley et al. (US 6,415,299)** as applied to Claim 20 above, and further in view of **Carey et al. (US 6,947,945)**.

As per **Claim 21**, the rejection of **Claim 20** is incorporated; however, Hotti et al. and Baisley et al. do not disclose:

- wherein the database schema version management structure definition comprises an XML schema definition.

Carey et al. disclose:

- wherein the database schema version management structure definition comprises an XML schema definition *(see Column 1: 44-49, "XML schemas specify constraints on the structures and types of elements in an XML document." and "Other XML schema definitions are also being developed, such as XML Schema ... ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey et al. into the teaching of Hotti et al. to include wherein the database schema version management structure definition comprises an XML schema definition. The modification would be obvious because one of ordinary skill in the art would be motivated to specify constraints on the structures and types of elements in an XML document *(see Carey et al. – Column 2: 44-49)*.

As per **Claim 22**, the rejection of **Claim 21** is incorporated; however, Hotti et al. and Baisley et al. do not disclose:

- wherein the schema data is maintained in an XML file structured according to the XML schema definition.

Carey et al. disclose:

- wherein the schema data is maintained in an XML file structured according to the XML schema definition *(see Column 1: 30-32, "An alternative data format to the tables found in an RDBMS is XML, which is a tag language for describing documents.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Carey et al. into the teaching of Hotti et al. to include wherein the schema data is maintained in an XML file structured according to the XML schema definition. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize a future standard for information exchange between peer data stores, and between client visualization tools and data servers *(see Carey et al. – Column 2: 15-17)*.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / QC
April 24, 2007

WEI ZHEN
SUPERVISORY PATENT EXAMINER